eNav: Smartphone-based Energy Efficient Location Sensing for Low-Power Vehicular Navigation

Shaohan Hu[†], Lu Su[*], Shen Li[†], Shiguang Wang[†], Chenji Pan[†], Siyu Gu[†], Md Tanvir Al Amin[†], Hengchang Liu[‡], Suman Nath[§], Romit Roy Choudhury[†], and Tarek F. Abdelzaher[†]

[†]University of Illinois at Urbana-Champaign
[*]State University of New York at Buffalo
[‡]University of Science and Technology of China
[§]Microsoft Research Redmond
{shu17, shenli3, swang83, cpan8, siyugu2, maamin2, croy, zaher}@illinois.edu, lusu@buffalo.edu, hcliu@ustc.edu.cn, sumann@microsoft.com

Abstract

This paper presents eNav, a smartphone-based vehicular GPS navigation system that has an *energy-saving* location sensing mode capable of drastically reducing navigation energy needs. Traditional navigation systems sample the phone GPS at a fixed rate (usually around 1Hz), regardless of factors such as current vehicle speed and distance from the next navigation waypoint. This practice results in a large energy consumption and unnecessarily reduces the attainable length of a navigation session, if the phone is left unplugged. According to a survey we conducted of 500 drivers, more than 37% said they ran out of battery while using a phone for navigation, and as much as 91% said they would like to have a vehicular navigation application with an energy saving mode. To meet this need, eNav exploits on-board accelerometers for approximate location sensing when the vehicle is sufficiently far from the next navigation waypoint (or is stopped), while using actual GPS sampling only when an accurate estimate is needed. A user test-study of eNav shows that it reduces navigation energy consumption by around 80% without compromising navigation quality and user experience. The paper contributes to low-power location sensing in the context of phone-based vehicular navigation.

1 Introduction

This paper describes eNav, a smartphone-based GPS navigation system with a novel power-conserving mode. The system makes a contribution to low-power location sens-

ing in the context of vehicular navigation. The distinguishing feature of location sensing *in the context of vehicular navigation* is that the location estimate does *not* have to be accurate at all times for navigation errors to be prevented. Rather, it is allowed to get inaccurate in certain situations (e.g., when the vehicle is far away from the next navigation waypoint[1]), while needing to remain accurate in others (e.g., when a waypoint is near). Hence, energy can be saved via an adaptive mechanism that keeps the location estimation error below a bound that changes depending on the current situation. The mechanism judiciously switches between a cheap inaccurate estimation mode and an expensive accurate one, depending on the allowed location estimation error at the current time.

The motivation for this paper comes from the observation that smartphones have become popular means for navigation in vehicles. Dedicated GPS navigation devices, such as Garmin, see a continued decline in market share [8], whereas integrated dashboard systems are still an expensive option, compared to smartphone applications. Unfortunately, the GPS module is one of the most power-hungry components on phones [20, 22, 25, 26, 29]. It may deplete batteries within hours (or less when the phone is not fully charged), running the risk of navigation loss while driving.[2] The above observations beg the question: would an energy-saving mode be a useful addition to current phone-based GPS navigation applications used by drivers? If so, how should it be implemented? In this paper, we first show results of a user-study that answers the first question in the affirmative. We then present the design, implementation and evaluation of such a service, demonstrating significant energy savings.

Briefly, eNav allows the user to enter or exit an energy saving navigation mode at will. In that mode, two mechanisms are employed that reduce energy consumption; *adaptive GPS sampling* and *screen saving*. Adaptive GPS sampling refers to substituting actual GPS positioning with dead reckoning using cheaper sensors, whenever such a substitution is deemed safe. The substitution is deemed safe as long as it cannot lead to a navigation error (for example, it is safe when the vehicle is sufficiently far from the next navigation waypoint). Screen saving refers to turning the screen off, ostensibly to save energy, but in reality to mask the fact that location estimation is inaccurate at certain parts of the route. As a waypoint approaches, the allowable location estimation error shrinks, GPS sampling restarts, and voice navigation alerts the driver to needed actions, making it look as if location estimation was accurate all along. Both of the above mechanisms contribute to improved energy-efficiency, while keeping location estimation inaccuracy transparent to the driver. Importantly, the evaluation shows that adaptive GPS sampling (which is the new contribution in this paper) significantly increases savings over screen saving alone (which we use as a baseline for comparison).

To assess the need for eNav, we conducted a user survey[3] using CrowdFlower.com [7], between Nov 20th 2013 and Jan 5th 2014, asking the participants about their preferences regarding phone-based vehicular navigation. The survey was terminated upon receiving 500 valid responses. Survey results, detailed later in the

[1] We define a *navigation waypoint* as a point on the route where the driver must take an action, such as making a turn or taking an exit.

[2] Indeed, the "real" motivation for this paper arose when an author nearly missed a flight, after his phone battery ran out while navigating to the airport.

[3] Under IRB Approval #14266.

paper, suggested three important observations. First, the intuition that smartphones are widely used for vehicular GPS navigation was corroborated by survey data. Second, while the car has ample power, the great majority of drivers did indeed welcome a power-saving navigation mode on the phone to use while driving. Finally, turning off the phone screen in the energy saving navigation mode was deemed acceptable by most drivers. These results complete the motivation for the new service.

The low-power location sensing approach used by eNav differs from prior location sensing work in two ways. Briefly, most recent phone-based systems [5, 35] that use dead-reckoning for localization are for pedestrians. In contrast, we specifically target dead-reckoning in *vehicles*. This makes the problem different. For example, we cannot rely on counting of walking steps to determine distance and orientation. To the best of our knowledge, eNav is the first smartphone-based system that exploits the phone's on-board motion sensors to achieve energy-efficient *vehicular* navigation. Second, the work differs from earlier dedicated systems [1, 33] where the sensors are rigidly placed in the environment. This is because in an everyday driving scenario, we cannot expect phones to be placed at a fixed location and orientation in the car. Instead, eNav features an innovative algorithms for extracting direction, velocity, and acceleration of car motion without assumptions or phone orientation.

The system presented in this paper has been implemented and empirically evaluated via a deployment study involving 33 external (non-author) drivers who used the service. Participants were given randomly chosen destinations to navigate to using our prototype eNav implementation. We made sure that the participants were not familiar with the destinations they were given, and hence had to rely on the navigator. A total of over 3,000 km of navigation trips were logged, spanning various road, traffic and weather conditions. Results of the deployment study showed that our energy-efficient navigation system achieves over 80% energy saving compared to standard GPS, without missing navigation waypoints.

The rest of the paper is organized as follows. A study assessing the need for eNav is presented in Section 2. We give a high level overview of our main contributions in Section 3. Section 4 and Section 5 describe the two main contributions; location estimation and adaptive GPS sampling. We present the implementation of eNav in Section 6 and evaluate it in Section 7. We discuss limitations of the approach in Section 8, present related work in Section 9 and, finally, conclude in Section 10.

2 Motivation

The idea for eNav originated from the authors' own bad personal experience with loss of navigation while driving. To motivate saving energy in a phone-based vehicular navigation system, however, the authors needed to answer three questions: Do (other) drivers commonly use phone-based navigation in vehicles, as opposed to other in-vehicle navigation options? Would saving phone energy while navigating be useful to them, despite the abundance of energy in a car? Finally, how do drivers feel about falling back on voice navigation, instead of visual cues, in the energy saving mode? (The latter is a side-effect of needing to mask location estimation inaccuracy in our service.)

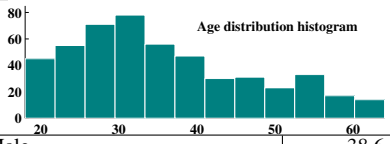
Questions	Choices	Responses (%)
Age?		
Gender?	a) Male b) Female	38.6 61.4
How often do you drive a car?	a) Very rarely b) Once or twice a week c) Everyday	9.6 21.8 68.6
Why do you drive? (Check all that apply)	a) To go to school b) To go to work c) To go shopping d) For occasional entertainment e) For long distance travel	23.2 65.2 84.4 69.6 46.0
How long is your average drive?	a) 15 minutes or less b) About half an hour c) About one hour d) More than an hour	23.8 52.2 19.0 5.0
Did you ever use a GPS navigation system in your car?	a) Yes d) No	80.4 19.6
Do you keep a phone charger or adaptor to enable charging your phone while in the car?	a) Yes, always b) Most of the time c) Sometimes d) Very rarely	15.8 20.2 29.8 34.2
Did you ever use a phone-based GPS navigation system in your car (or a rental car)?	a) Yes b) No	73.0 (59.4) 27.0 (40.6)
If you answered Yes to either of the above, do you usually plug-in your phone to the car charger when using phone navigation in a car?	a) No b) Sometimes c) Most of the time d) Yes, always	28.0 36.0 20.2 15.8
Did you ever run out of battery on the phone while using the phone for navigation?	a) Yes b) No	37.4 62.6
Imagine a phone-based GPS navigation system with an optional battery-saving mode that turns off the screen when it is not needed (e.g., when you are not moving or far from your next turn-point). Voice navigation will continue at all times. How useful would you find this battery saving feature?	a) I would not find it useful b) I'd use it when my battery is low c) I'd make it the default mode for my navigator	9.0 61.0 30.0
Which do you think is more important for vehicular GPS navigation: screen display, or turn-by-turn voice guidance?	a) Screen display b) Voice guidance c) Either	25.4 45.6 29.0
If during navigation your phone battery is running low, and turning off your screen can save significant amount of battery life, would you be willing to do turn off the screen and rely just on turn-by-turn voice guidance?	a) Yes b) No	81.9 18.2

Table 1: Survey questions and responses.

To answer these questions, we carried out a nation-wide online survey (in the United States) using CrowdFlower.com [7]. All questions were multiple-choice. In order to filter out less reliable responses (possibly due to respondents not paying enough attention or simply providing random answers), each survey contained 5 randomly-placed repeated questions with reordered choices. We only considered responses that showed consistency across all the repeated questions. We ran our survey until 500 valid responses were received. Survey questions and basic answer statistics are shown in Table 1. Each participant was paid $0.5 (US) for completing the survey, which is consistent with prevailing compensation rates on CrowdFlower. The survey engine had meachanisms to prevent repeated entries by the same user.

Demographics: Survey respondents covered 385 cities in 48 different states in the US, of whom 38.6% were male and 61.4% female, ranging from 18 to 64 years of age (mean 35.9 and standard deviation 11.7). Most respondents were frequent drivers. Specifically, 68.6% said they drove every day, 21.8% said they drove once or twice a week, whereas only 9.6% said they drove rarely. Most of the driving was associated with shopping (84.4%), occasional entertainment (69.6%), work commute (65.2%), and long distance travel (46%), in that order. More than 75% said their average commute was at least 15 minutes.

Prevalence of phone-based GPS: The majority of respondents said they used phone-based GPS in the car. More specifically, 80.4% said they used GPS in a car, and 73% said they used a phone-based GPS in a car. More than 59.4% also used a phone-based GPS in a rental car (which often implies being away on a trip and in need of navigation), and as much as 37.4% said that they experienced running out of phone battery while using the phone for navigation.

Interest in energy-saving navigation (eNav): When asked whether they would find an energy saving navigator useful, 91% of the respondents said they would like to have an energy-efficient phone navigation application, of whom, roughly 2/3 said they would choose the energy saving mode when their phones are running low on battery, while 1/3 said they would make it the default mode regardless of the phone battery status, which is an even stronger endorsement.

We were especially interested in finding out the demographics correlated with interest in eNav. Taking survey responses as ordinal values, we computed the correlations between these responses and interest in eNav. *Statistically significant* positive correlations were found between interest in eNav and each of (i) being a frequent driver, (ii) using GPS, (iii) using GPS on a phone, and (iv) running out of battery while navigating. This means that individuals with more driving and GPS usage experience are precisely those who liked eNav more. Not surprisingly, individuals who suffered from a navigation outage thanks to battery depletion also appreciated the service more. What was more surprising was that interest in eNav was also found to have a statistically significant positive correlation with the frequency of using a phone charger in the car. This suggests that individuals who use the charger more frequently do not particularly like having to do so, and are thus more appreciative of eNav.

Voice versus screen: Finally, 75% of the respondents either considered voice to be more important than visual cues for navigation, or were fine with either mode. Moreover, 81.8% said they would be willing to rely on voice navigation in the car if their phone battery was running low. There was a statistically significant correlation between

accepting voice navigation (in lieu of the screen) and liking eNav, as well as being a frequent driver.

We summarise three key observations from the above results. First, the intuition that smartphones are widely used for vehicular GPS navigation is corroborated by survey data. Second, while the car has ample power, the great majority of drivers do indeed welcome a power-saving navigation mode on the phone. Finally, turning off the phone screen in the energy saving navigation mode is acceptable to most drivers. The latter observation was important to us. While the contribution of eNav lies in its approximate low-power location sensing, and not in saving screen power, eNav *has to* turn off the screen in order to mask the fact that its location estimate gets inaccurate (specifically, when the car is far from the next navigation waypoint). Hence, it was important to determine whether drivers will accept that. The above results complete our motivation for eNav.

3 Overall Contributions

The goal of a GPS navigator is simple: instruct drivers to follow a specified route and minimize navigation errors. In order to build an energy-efficient navigation system, GPS usage needs to be reduced by replacing GPS with less expensive sensors while meeting the above goal. This naturally leads to the following two questions: (i) How to estimate the car's location when the GPS is off? (ii) When to turn GPS back on to prevent navigation errors? These questions yield the main contributions of our work as discussed below.

How to estimate car's location with GPS off? We recognize that the phone's on-board motion sensors (e.g., accelerometers) are a natural candidate for estimating location when GPS is off: They consume only a tiny amount of energy (0.0488mW at 10Hz, according to our measurement) compared to the GPS (150mW at 1Hz) and can provide inertial motion readings, which can be used to estimate locations by carrying out dead-reckoning. Network-based localization (cellular triangulation and WiFi SSID signatures) has also been studied [16, 17]. We found from our experiments using Android's network-based localization implementation that its highest sampling rate was only at about 1 sample per 20s, which we considered too low for navigation. Therefore, we focus on the phones' on-board motion sensors for our system. Our first contribution thus lies in approximate *low-power vehicular location sensing* (namely, displacement measurement in the direction of car motion).

When to turn GPS on? To answer this question, one might propose the naïve approach of simply lowering the GPS sampling rate during navigation while in the power saving mode. This approach *will* reduce the energy consumption, but at the expense of uniformly lowered localization accuracy and degraded navigation quality. As later shown in Section 7.1.2, due to the non-linearity of GPS energy consumption, saving about 80% energy (which our eNav system achieves) would dictate that the GPS sampling period of a traditional navigation application be increased from 1s to a whopping 83s, which according to our experiments would cause the user to miss about 83.1% of all waypoints during navigation! We, on the other hand, take an adaptive approach derived from the simple intuition that high localization accuracy is needed

when the car is close to a waypoint, but not when the car is still far away. Thus, given a rough estimate of the car's location, we can adaptively decide when to sample the GPS next. *Adaptive GPS sampling* is our second main contribution. A related important aspect of navigation is to handle driver errors, which we do as an enhancement to the basic adaptive GPS sampling algorithm.

The above contributions are discussed in the following sections, respectively.

4 Low-Power Location Sensing

To estimate location when GPS is turned off, we exploit accelerometers, while resorting to GPS sampling only when the estimated location error exceeds a bound. Accelerometers are useful since the displacement vector can be obtained as the double-integral of the acceleration vector, which is known as *dead reckoning*. Two questions arise in that context. First, given a phone's acceleration measurements and assuming that the phone orientation is perfectly aligned with the direction of motion, how accurate is location estimation? This is the best-case scenario and the answer determines the feasibility of using phone accelerometers for vehicle location estimation in the first place. Second, how would one estimate acceleration in the direction of motion given an arbitrary and unknown phone orientation in practice? It represents the typical usecase scenario. These questions are discussed in the following subsections, respectively.

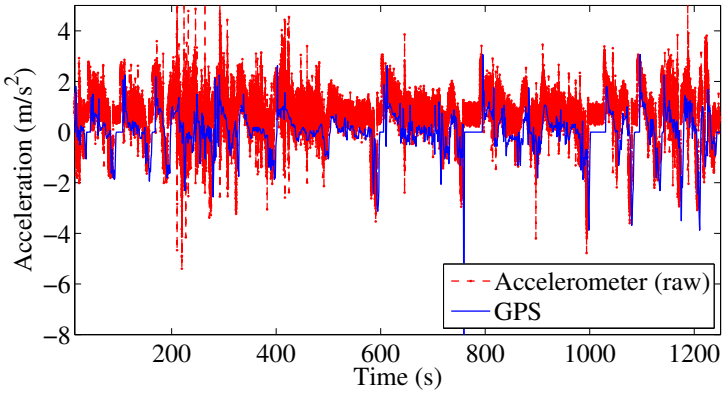
4.1 A Best Case Scenario

In order to understand whether location estimation is technically doable using the quality of accelerometers available on a phone, we first conduct an ideal experiment where the phone is carefully mounted in the car, such that a particular accelerometer axis is perfectly aligned with the car's direction of motion. We then drive on a horizontal road to eliminate interference from the gravity axis.

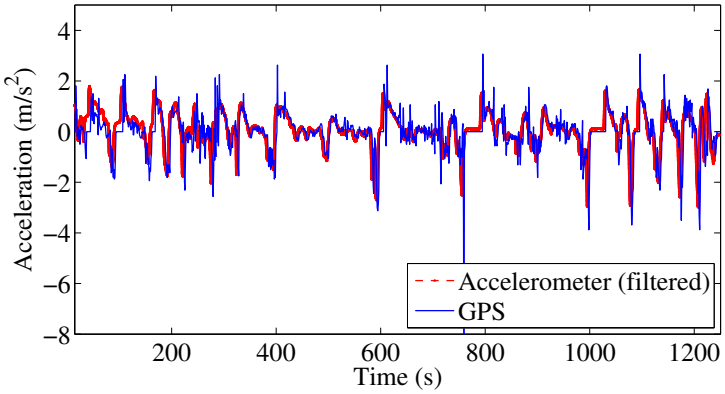
In this experiment, a Galaxy Nexus Android phone was used. We drove the car on a predefined route, during which the phone continuously collected and logged onboard accelerometer (at 10Hz) and GPS (at 1Hz) readings. Since our goal is to mimic GPS, the collected GPS trace was treated as the ground-truth, offering position and speed measurements (from which acceleration was also computed). These measurements were compared against the phone's accelerometer readings (which were also integrated to obtain speed and displacement). We repeated the experiment with 5 different driver-car pairs on 3 different routes, each consisting of around 10 different segments of various lengths. A total of 200 km of driving data was collected.

Fig. 1 shows an example trace, where a car's acceleration measured from the phone's accelerometer is compared to that computed from the phone's GPS readings. In Fig. 1(a), it can be seen that the raw accelerometer readings generally follow the trend of the car ground-truth acceleration and the noise in raw accelerometer data is effectively removed by a simple low-pass filter, as seen in Fig. 1(b). The cut-off frequency of the low-pass filter was set to 0.1Hz, determined experimentally, indicating that the car's acceleration signal is in the low frequency range (below 0.1Hz), whereas

the higher-frequency components are attributed to sources like road bumpiness, the car's vibration, and the sensor's intrinsic noise.



(a) Raw accelerometer data compared to the acceleration computed from GPS trace



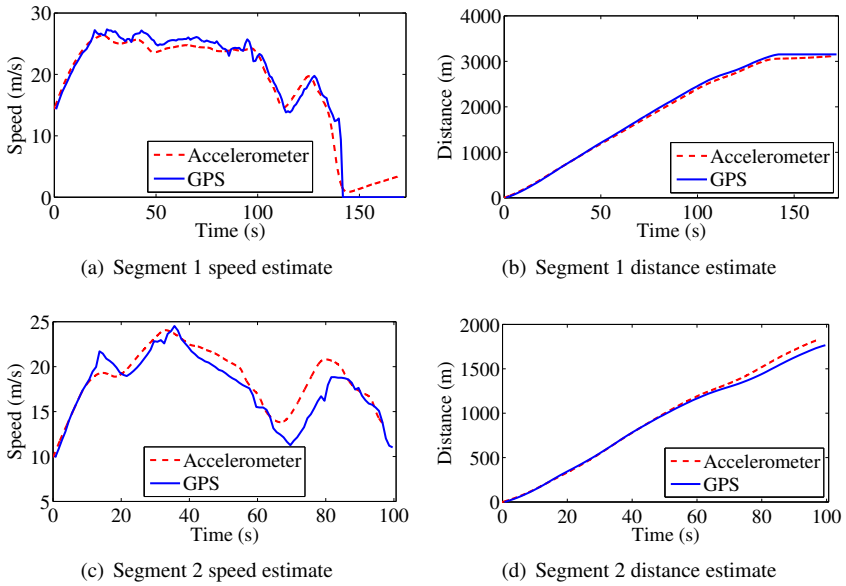
(b) Filtered accelerometer data compared to the acceleration computed from GPS trace

Figure 1: Car acceleration estimation using the phone's accelerometer data

Given the acceleration estimates, we integrate them over time obtaining the corresponding speed and distance estimation, as illustrated in Fig. 2 for two sample road segments. The distribution of the distance estimation error (per unit of route segment length) computed over the entire 200 km of driving data is shown in Fig. 3. The distribution has an average that is well below 1% (representing the mean distance estimation error), with a standard deviation of 0.02. Hence we deem phone-based dead-reckoning feasible, albeit imperfect. Next, we explore arbitrary and unknown phone placement.

4.2 Practical Acceleration Estimation

To perform position estimation of a car on a road, using a phone whose orientation is arbitrary and unknown, we need to solve two problems: (i) extract the acceleration



(a) Segment 1 speed estimate

(b) Segment 1 distance estimate

(c) Segment 2 speed estimate

(d) Segment 2 distance estimate

Figure 2: Car driving speed and distance estimation using the phone's accelerometer data

measured along the car's direction of motion by using the phone's local sensor readings without prior knowledge of the phone's orientation relative to the car, and (ii) remove the effect of gravity component from the acceleration measurements along the car's direction of motion (when the car is not driven on horizontal roads). The Android phones that we used implement proprietary solutions, called Sensor Fusion [21], to solve these problems. We explore their accuracy in Section 4.2.1. Finding them inaccurate, we implement our own solution, called *Principal Motion Estimation* (PME), which we discuss in detail in Section 4.2.2.

4.2.1 Using Phone's Proprietary Sensor Fusion

Implemented by the phone's sensor vendor [12], sensor fusion is a set of proprietary algorithms that combine raw physical accelerometer, gyroscope, and magnetic field sensor readings to correct each other's errors and produce an array of *virtual* sensor readings, including gravity, linear acceleration, and the rotation vector [21]. The gravity and linear acceleration virtual sensors are supposed to separate the earth gravity effect and the phone's actual acceleration given the raw accelerometer readings. The rotation vector contains the rotation matrix that can be used to rotate the phone sensor readings to the earth's global coordinate system. Therefore, given these fusion virtual sensor readings, we should be able to compute the phone's acceleration in the earth coordinates, and in turn estimate them in the car's direction of motion.

To test whether or not the built-in proprietory sensor fusion algorithm could solve our problems, we carried out a set of experiments similar to the previous one, discussed

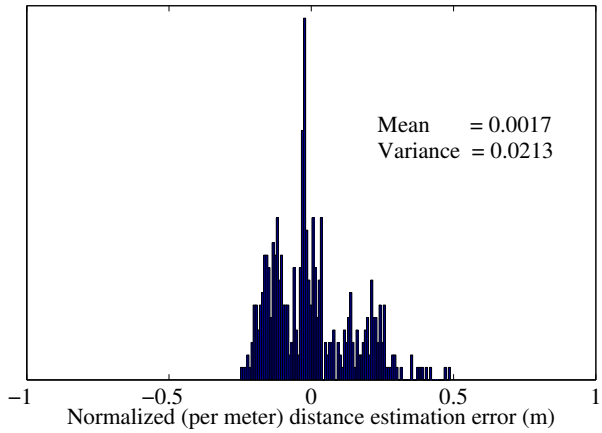


Figure 3: Distribution of the normalized distance estimation error (segment estimation error/segment length) when performing dead-reckoning using a phone's accelerometer

in Section 4.1, with the only differences being that (i) routes were selected that vary in conditions such as slope, bumpiness, and traffic, and (ii) multiple phones were put in each car with different random placements and orientations.

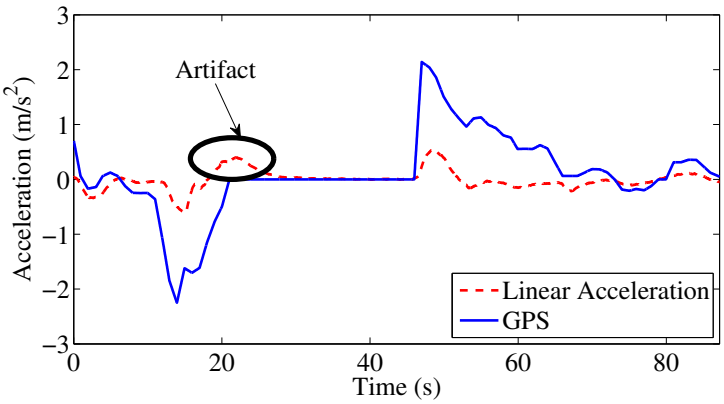


Figure 4: Car motion estimation using phone's proprietary Sensor Fusion

Fig. 4 shows a representative example of acceleration estimation results. As seen, the car's acceleration is greatly attenuated as estimated using sensor fusion. We also notice unexplained artifacts created in the virtual sensor data. To quantify the estimation quality, we, again, computed the normalized distance estimation error histograms for all driving data. The results are shown in Fig. 5, for three different phone placements; namely, resting on the seat, mounted on the dashboard, and placed in pocket. The corresponding distribution statistics are summarized in Table 2. We see that when

	Mean		Variance	
Placement	PME	Fusion	PME	Fusion
Pocket	-0.0906	-0.3437	0.0685	0.1100
Dashboard	0.0272	-0.3873	0.0269	0.3755
Seat	0.0063	-0.4002	0.0282	0.0748

Table 2: Comparison of the normalized distance estimation error distribution statistics: PME *vs* sensor fusion

using the proprietory sensor fusion algorithm, the resulting error has a large bias. This is bad for dead-reckoning as it then accumulates a large error over time. In this case, Table 2 shows that vendor's fusion *underestimates* distance traveled by the car by about 34% to 40% on average. For comparison, our proposed PME method, discussed in Section 4.2.2, produces average normalized errors below 10% for pocket placement and in the 0-3% range for other placements of the phone, which is an order of magnitude improvement. It can also be seen from Table 2 that the sensor fusion-based method results in a higher variance for normalized distance estimation errors when compared to our PME method.

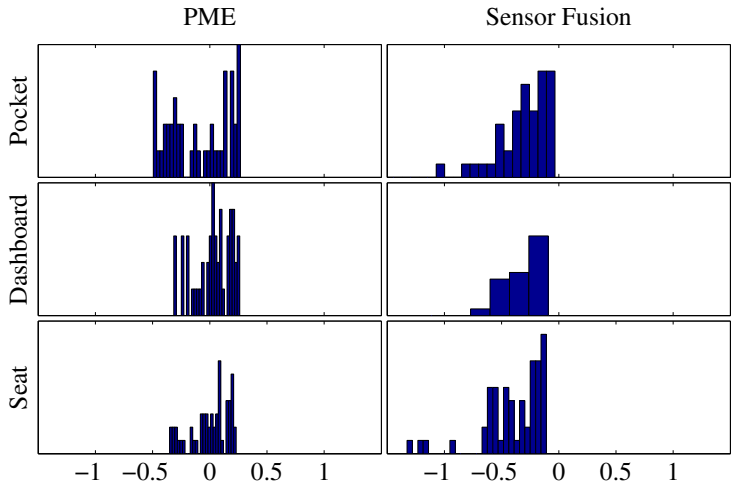
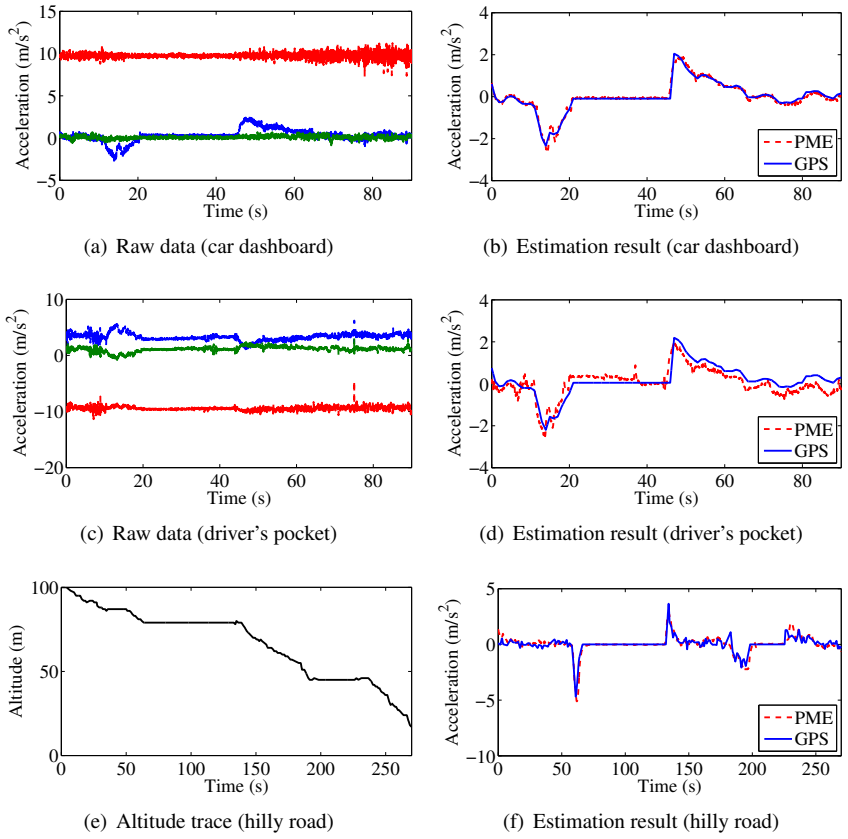


Figure 5: Normalized distance estimation error histograms for our proposed PME method *vs* using the phone's proprietary sensor fusion

This set of experiments suggests that the phone's proprietary sensor fusion is not suitable for our target vehicular setting. We speculate that the vendor designed and implemented these sensor fusion algorithms for a different goal; namely, to optimize performance for everyday user-phone interactions. Users mostly interact with their phones outside of moving vehicles. Hence, it is likely that the vendor's algorithms are not optimized for our usecase.



(a) Raw data (car dashboard)

(b) Estimation result (car dashboard)

(c) Raw data (driver's pocket)

(d) Estimation result (driver's pocket)

(e) Altitude trace (hilly road)

(f) Estimation result (hilly road)

Figure 6: Principal Motion Estimation result demonstrations

4.2.2 Principal Motion Estimation

We now discuss our novel Principal Motion Estimation (PME) method. PME is based on the technique of Principal Component Analysis (PCA) [13]. The goal of PCA is to find a new set of components that better captures the variability (i.e., variance) of the data. Specifically, the first component direction is chosen to capture as much variability of the data as possible. Orthogonal to the first component, the second is chosen to capture as much of the remaining variability as possible, and so on. In our case, we apply PCA on the phone's 3-axis accelerometer data. The first component derived by PCA captures the largest variability of the car's acceleration, which intuitively should correspond to the car's driving. The second and third components capture the remaining acceleration variability of the car's motion, and might correspond to the directions of the car making left and right turns, and vertical movements (e.g., on bumpy roads).

Mathematically, let $\mathbf{D}$ be a raw accelerometer data trace with its mean subtracted, which serves as the input of PCA. The output would be a matrix $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$, in which each column vector is an eigenvector of $\mathbf{D}$'s covariance matrix. The acceleration

component corresponding to the car's driving direction is thus given by $\mathbf{Du}_1$. For estimating distance traveled, we do not need to worry about the components along the car's other two axes. Please also note that the mean of the data is subtracted in order to remove the gravity's effect on the accelerometer measurements, as for a reasonably long time interval, the average acceleration along all axes of the car should be close to 0.

The vector $\mathbf{u}_1$ determines the phone orientation with respect to the principal direction of motion. Once computed, it can be used for the remainder of the trip, or until the phone is bumped, changing its orientation. Our current prototype does not detect such bumps, but in principle they can be detected using techniques such as Jigsaw [20] and Nericell [24], in which case we simply re-run $\mathbf{u}_1$ estimation.

Note that, the principal component produced by the PCA has ambiguous signs. Since cars spend most of their time moving forward, we simply assume that the initial acceleration after any stop (i.e., period of zero speed) is in the forward motion direction. We also considered using Nericell [24], which introduced another method for determining accelerometer orientation in cars. However, it requires the sampling of GPS, which introduces extra energy costs and hence did not serve our purpose of low power navigation.

Next, we show examples of PME-based estimation results. First, consider the same driving segment referred to in Fig. 4, in the context of discussing the vendor's fusion algorithm, which did not produce an accurate match. Fig. 6(a) and 6(b) show the 3-axis raw accelerometer data and the PME-estimated car acceleration as compared to the ground-truth, when the phone is on the dashboard. Fig. 6(c) and 6(d)) repeat the same when the phone is in the pocket. A good correspondence with ground truth is observed in acceleration measurements.

Next, we consider a trace from a hilly road. Fig. 6(e) shows the altitude trace of the car as it was driven on that road. Since the road was not horizontal, the gravity had a non-zero component in the car's driving direction. Fig. 6(f) reports the car's acceleration estimate produced on that road by our PME method, showing good correspondence with ground truth.

To offer a more quantitative measure of accuracy, we use the raw accelerometer data collected from experiments involving driving trips on routes that spanned various road (e.g. horizontal, hilly, rural, urban) and weather (e.g. sunny, rainy, snowy) conditions, at different times of days, under varying traffic conditions. The experiment was repeated with 5 different driver-car pairs on 3 different complex routes, each consisting of around 20 different segments of various lengths. A total of 400 km of driving data was collected. The produced distance estimation error distribution statistics are shown in Table 2. The table demonstrates that the average estimation error is small enough that the approach is sufficient for location estimation in between GPS samples. In the next section, we describe an algorithm that leverages the above results for adaptively sampling GPS depending on the current accumulated distance error estimate, and the position of the next navigation waypoint.

5 Adaptive GPS Sampling

The idea of adaptive GPS sampling is simple. First, integrate the measured acceleration twice to compute a displacement estimate along the current navigation segment. Second, from the standard deviation in acceleration, compute the standard deviation in displacement. Third compute a confidence interval (confidence window) around the current displacement estimate using the obtained standard deviation in displacement. We use a window that extends two standard deviations around the mean, which corresponds to a 97% confidence interval. Finally, if the next navigation waypoint is at least distance L outside that window, then we know that we are at least distance L from the waypoint (with the chosen confidence, which in this case is 97%). Hence, no GPS sampling is needed. Otherwise, a GPS sample is taken and the accumulated location error is reset to zero. The configurable threshold L can be chosen to correspond to a comfortable warning distance for the driver. The chosen confidence interval is a tradeoff between energy savings (keeping GPS off longer) and false negatives (missing a waypoint). The adaptive GPS sampling algorithm is detailed below.

5.1 Basic Error and Location Estimation

We discretize the continuous time into slots (each of duration ΔT), and assign an index for each time slot with the first slot index being 0. Within each time slot, the speed and acceleration are considered constant. Let $\hat{a}_k$ be the reading of the accelerometer at time k. Further more, let $\hat{v}_k$ and $\hat{s}_k$ be the estimated speed and displacement at time k, obtained by numeric integration of the acceleration timeseries. We measure displacement from the last known GPS sample. Hence, when GPS is sampled, we initialize the integration, setting $\hat{s}_0 = 0$ and $\hat{v}_0$ to the velocity value obtained from GPS.

When GPS is not available, we measure acceleration and do the numeric integration. Thus, $\hat{v}_k = \hat{v}_{k-1} + \Delta T \hat{a}_k$ and $\hat{s}_k = \hat{s}_{k-1} + \Delta T \hat{v}_k$.

Let v_k and s_k denote the actual speed and displacement at time, k, and define error e_k^a, e_k^v, e_k^s as the acceleration, speed and position errors at time k. Recursively expanding the numeric integration and replacing the estimated variables by the sum of ground truth plus error, we eventually get at time k:

$$\hat{v}_k = v_k + \Delta T \sum_{0<i\leq k} e_i^a \quad (1)$$

$$\hat{s}_k = s_k + (\Delta T)^2 \sum_{0<j\leq k} \sum_{0<i\leq j} e_i^a \quad (2)$$

In other words, the speed estimation error at time k is the sum of k terms of the acceleration estimation error multiplied by ΔT and the displacement estimation error at time k is the sum of (roughly) $k^2/2$ terms of the acceleration estimation error multiplied by $(\Delta T)^2$. Assuming that the errors in acceleration estimation are independent identically distributed random variables following a Gaussian distribution $\mathcal{N}(0, \sigma^2)$, we can compute the means and standard deviations of the speed estimation error and

displacement estimation error by invoking the law of sums of random variables. It simply states that their means and variances add up. Hence, $e^v \sim \mathcal{N}(0, \sigma_v^2)$, where $\sigma_v = \Delta T\sqrt{k}\,\sigma$, and $e^s \sim \mathcal{N}(0, \sigma_s^2)$, where $\sigma_s = (\Delta T)^2 \frac{k}{\sqrt{2}}\,\sigma$.

With the variance of displacement computed, we compute the car's confidence interval in its estimated position. We use a 97% confidence interval, which roughly corresponds to $2\sigma_s$ around the mean. We require that the next waypoint be at least L seconds outside that interval, where L is the driver's warning time. In other words, it should be outside that interval by a distance equal to current speed times L. If so, no GPS sampling is needed. Otherwise, we obtain the accurate location of the car by sampling the GPS.

As a practical consideration, when performing numeric untegration to compute current speed, we bound it between zero and 15 mph above speed limit, as we consider speeds outside that box to be erroneous. This bounding prevents errors from acumulating, causing the speed to reach unrealistic numbers. For our service, we found a web resource, Wikispeedia [37], that hosts publicly available crowd-sourced road speed-limit data. We were able to crawl the data covering our regions, hence implementing the above feature.

5.2 Enhancements to Location Estimation

In this section we discuss enhancements to location estimation that we developed for improving estimation results during navigation.

5.2.1 Car-Idle Detection

As previously mentioned, being able to detect car idling could help reset both the acceleration and speed estimations and prevent unnecessary growth of the corresponding distance estimation errors. Treating the detection task as a binary classification problem (where for each time slot we classify the accelerometer data as reflecting the car being idle or not idle), we initially experimented with a simple threshold-based method, for which we just took the magnitude of the raw 3-axis accelerometer data for each time slot (1s window) and compared its $mean$ to the learned threshold. Intuitively, when the car is idle, the magnitude of acceleration should be around 9.81, which is gravity, hence suggesting a threshold based approach. The approach yielded about 90% accuracy, with occasional misclassifications. This is because a good car on a good road offers a smooth enough ride that the accelerometer may not distinguish between being still and moving at constant velocity in a straight line.

Hence, in addition to the $mean$ acceleration magnitude, we computed the min, max, and $standard\ deviation$ to form a $4D$ feature vector for the classification task. Note that, for training, we simply labeled the time slots using speed readings of the corresponding GPS trace.

We experimented with several classification algorithms, using 10-fold cross-validation to compare their accuracy, defined as the average percentage of correctly labeled time slots among all slots tested. Our experiments show that the decision tree algorithm [30] achieves near perfect classification, as shown in Table 3. Therefore, we

use a decision tree classifier in our final system design to detect idle time. The classifier is trained on the car's own data.

Classification Algorithm	Car-Idle (%)	Car-Turning (%)
Decision Tree	99.80	98.89
Support Vector Machine	96.35	69.48
Naive Bayes	98.17	63.63

Table 3: Car-idle and car-turning detection accuracy comparisons using various classification algorithms (10-fold cross-validation)

5.2.2 Car-Turning Detection

One other event we exploit is when a car turns. Combined with road intersection information, a turn gives us the opportunity to pinpoint a car's location without needing to sample the GPS. The intuition is straightforward. Whenever the car makes a turn, we check to see how many road intersections exist within the current location confidence window. If there is none or more than one, we are unable to determine accurate location of the car, so we sample GPS (to start a new segment). If there is only one, however, we can pinpoint the car as being at that intersection, without sampling the GPS. Road intersection information is obtained by processing the OpenStreetMap [27] (OSM) data, where intersections are identified as OSM nodes shared by multiple OSM ways. We carry out the extraction offline, and store the resulting intersection data locally on phones for use during real-time navigation. A single intersection is just a pair of (*latitude, longitude*) floats, thus caching even a large number of intersections (in the broad vicinity of the car) would not take much phone storage space.

The detection of car-turning is modeled as a binary classification problem, similarly to idle-detection, and uses the same set of features with the only difference being that each time slot is now 5s long (as we observed from data traces that turnings usually lasted about 5s). For training, each time slot is automatically labeled using the GPS bearing trace. Again, we observed that the decision tree classifier gives the best performance among all. We initially used a gyroscope for this experiment, which yielded similar results as the accelerometer. However, as gyroscope's energy consumption is about two orders of magnitude greater than that of the accelerometer (as shown later in Section 7.1.1), we decided to use accelerometer for turning detection.

With turning detection enabled, eNav would suspend GPS sampling upon delivering the navigation notification to the user and relies on the to-be-detected turning motion to snap the car to the way-point, whose true location is known beforehand. Therefore, this approach reduces the sampling of GPS near waypoints, and introduces additional energy savings.

5.3 Deviation Detection & Handling

The handling of human mistakes is mission-critical for a navigation system. We assume the common case where users are honestly trying to follow navigation instructions, as

opposed to trying to defeat their navigator. Three different user error scenarios are addressed:

1. The user makes a turn too early. This means that after being notified about an upcoming waypoint, there is at least one more intersection before the actual waypoint intersection. eNav will detect the turn immediately, as described earlier, and then try to localize the turn, either via map-based localization (if it is the only intersection in the uncertainty window) or a GPS sample (if it is not). As a result, the turn is identified as wrong, the user is notified, GPS is sampled, and a new route is computed.

2. The user makes a turn too late. In this case, the user, after hearing the navigation notification, misses the waypoint and takes a subsequent intersection. The detection of this type of deviation is exactly the same as the previous scenario if the wrong intersection is not far away from the waypoint. If it is, then it becomes identical to the next scenario.

3. The user fails to make a turn and keeps driving (possibly because there is no nearby subsequent intersections after the missed waypoint). In this case, eNav will keep updating the possible location range for the car to the point where the location confidence window moves beyond the waypoint intersection, at which point eNav recognizes that the user has likely missed the waypoint, and thus re-localizes by sampling the GPS.

6 Implementation

Our eNav prototype application was implemented from scratch to run on Android phones. When a user enables the energy-saving mode during navigation, adaptive GPS sampling is enabled and car localization is allowed to become inaccurate between waypoints. The phone screen will also turn off. As discussed previously, according to our survey results, the majority of people find it acceptable to rely on voice guidance to conserve phone battery.

We also made it such that the user can, at any time, pause the energy-saving mode by waking up the phone screen, at which point eNav will restore GPS sampling, and present to the user their accurate location, masking the fact that location was ever inaccurate. In our current prototype implementation, this interaction is done via user pushing the power button. More convenient interfaces, such as voice, are likely doable but are beyond the scope of the intended contribution of this paper.

Next, we discuss eNav's energy-efficient navigation flow, illustrated in Fig. 7. Nodes marked as Ox's and Dx's correspond to the basic operations and decisions, and eDx's are the enhancement components.

6.1 Basic Navigation

At the beginning of the trip, the GPS and accelerometer are both sampled until the car has gotten on the road and been driven for a short duration (empirically 1 min

is enough). Data collected during this phase is used for initializing the various eNav models: the PME principal component vector, and the acceleration estimation error distributions. Then, eNav's energy-efficient navigation can kick in whenever the user decides to turn it on.

We first describe how the basic navigation flow works. In real-time, for each time slot, the car's principal motion is computed from accelerometer data (O1), and the speed and location estimation of the car are updated (O2). The estimated speed and the displacement confidence interval are then used to estimate distance to the next waypoint, which is then translated into time. If this time is smaller than a threshold (D1), GPS is sampled to get the accurate speed and location information (O3), which in turn is used to compute the time it takes to reach the next waypoint under the current GPS speed. If this time is again smaller than a threshold (D4), then the car actually *is* close to the next waypoint. In this case, eNav notifies the user about the upcoming waypoint (O5), and keeps sampling GPS continuously until the user passes through the waypoint (unless the turn-detection enhancement is used). Otherwise, the car is still far away, no special action is taken.

If the user fails to follow navigation instructions and drives past the waypoint (D3), deviation detection eventually fires, energy saving stops, and eNav immediately recalculates a new route using the car's current location (O4). Finally, eNav notifies the user and ends the navigation (O6) upon reaching the final destination (D2).

As mentioned, two threshold values are used in the navigation flow. We call the one in D4 the *critical notification time*. This controls how far ahead should the user be notified about the upcoming waypoint. If it is too high (e.g., "Turn right in 5 min after 10 km"), the user will probably have already forgotten about the notification by the time s/he actually reaches the waypoint. If it is too low, the user will likely not have enough time to react. By testing various commercial navigation applications/devices and interviewing our test users, we decided on the value 10s, which users can also adjust to better fit their personal preferences. Please note that a navigation application can provide multiple notifications to the user about the same waypoint. As long as a notification is delivered at or before the critical notification time, we consider that the navigator succeeded in announcing the waypoint. The other threshold, as used in D1 is then set to be the time for GPS to get a fix under the current situation plus the critical notification time.

6.2 Enhanced Navigation

To incorporate car-idle detection (eD1), the acceleration data of each time slot is used to classify whether the car is idle or not, as described in Section 5.2.1. If it is, then eNav sets the car's estimated acceleration and speed to be 0 and does not modify the location estimate. Otherwise, eNav follows the rest of the basic navigation flow.

When car-turning detection (eD2a) is enabled, eNav checks for car-turning motion for each time slot. Upon detection, eNav tries to get the accurate turning location using map intersection location information (eD2b) by checking if a unique intersection can be identified within the car's location confidence window. If yes, an accurate location fix is obtained without using the GPS; and if not, GPS is sampled to get the accurate location information. Then, the rest of the basic flow is followed, except that eNav

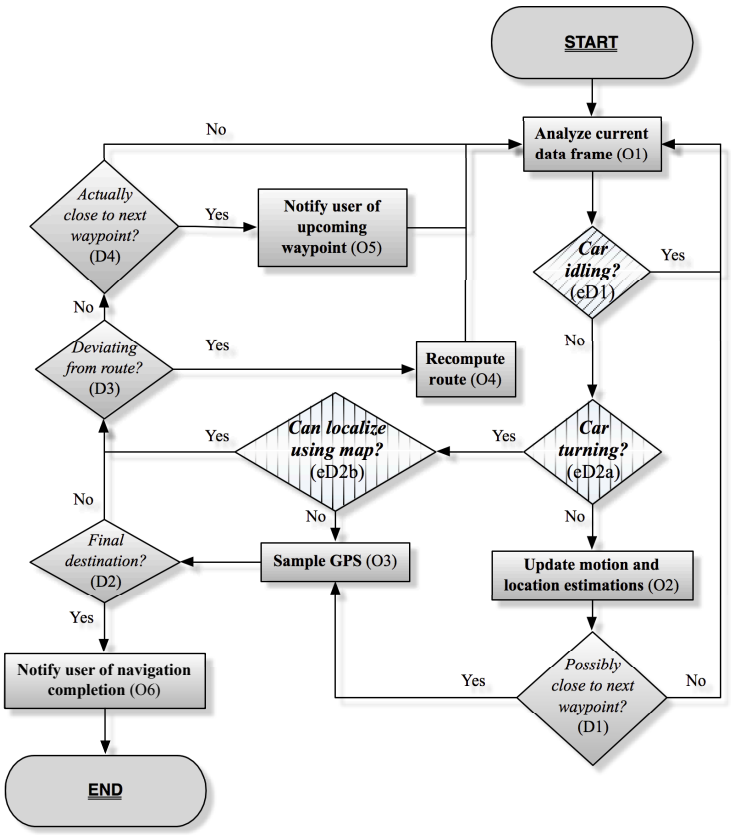


Figure 7: eNav's Navigation Flow

suspends all further GPS pull requests after notifying the user of the upcoming waypoint for the current segment when the turning detection module is enabled, and instead relies on turn detection to determine when the user has reached the waypoint.

7 Evaluation

During previous discussions on the various components of eNav, we have already shown corresponding evaluations specifically for those individual components (e.g., normalized distance estimation error, car-idle detection accuracy, etc.). Since our ultimate goal is to provide energy-efficient navigation, here we focus on evaluating eNav's energy efficiency and navigation quality.

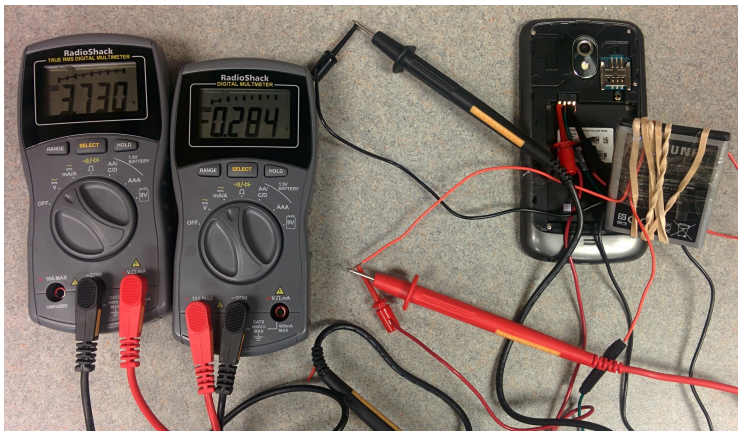


Figure 8: Configuration of phone energy consumption measurements. The left multimeter is measuring the voltage, and right one current.

7.1 Energy Models

We first describe in detail how we obtained the energy consumption models for the phone's on-board sensors (accelerometer and gyroscope), GPS module, and the computations that make up the eNav system. The models were needed to compute energy consumptions given driving data traces. Without such models, we would have to directly measure the energy consumption as eNav performs navigations as users drive, which would require physically connecting multimeters to the phone and battery, which would affect the phone's motion sensor readings, introducing extra error sources.

7.1.1 Phone's On-board Motion Sensors

The gross energy consumption of each of the on-board sensors was measured at 10Hz sampling rate. We used two multimeters for the concurrent measurements of voltage and current during phone operations. The voltage meter was connected in parallel to the phone's $+$ and $-$ connector pins, and the current meter in series between the battery's $+$ electrode and the phone's $+$ connector pin. The configuration is as illustrated in Fig. 8. To isolate the net energy consumption from the rest of the phone operations, measurements were made with as well as without actually sampling the sensor, the difference then gave us the net energy consumption. Each experiment was repeated 10 times and the average was taken. Also, all energy measurement experiments were carried out using the same phone within a span of 2 weeks, in order to avoid possible problems caused by different phones having different battery capacities or the same phone's battery capacity changing over a longer period of time. According to our measurements, accelerometer's and gyroscope's energy consumption rates were 0.0488mW and 2.14mW, respectively. This suggested not using the gyroscope.

7.1.2 Phone's GPS Module

For the phone's GPS module, a similar general approach was used to measure its power consumptions. However, due to its cold/warm/hot-start nature as also discussed in previous work [19, 23], GPS consumes energy at different rates under different sampling periods. Basically, the computation needed to acquire a single GPS fix increases as the time elapse since the previous fix lengthens, because the information from the previous fix becomes less useful and eventually expires. We thus needed to measure the GPS energy consumption at various sampling periods (from 1s to a couple of minutes, in our actual measurements) in order to be able to fit a quantitative model that we could use to compute the energy consumption of a particular GPS sampling trace. Our measurements are shown in Fig. 9. As seen, the per-sample energy consumption increases rapidly as the sampling period increases from 1s to about half a minute, and stays relatively stationary afterwards. We also notice that the general trend is gradual and smooth, showing no apparent GPS mode change with expensive wake-up cost. We therefore fit a simple continuous monotone function to model the GPS energy consumption, as shown by the curve superimposed over the measurements in Fig. 9. It is worth noting that, even though the per-sample energy consumption increases as the sampling period lengthens, it never gets to a point where, for a fixed sampling duration, sampling at a higher rate would consume less energy than at a lower one, as illustrated in Fig. 10.

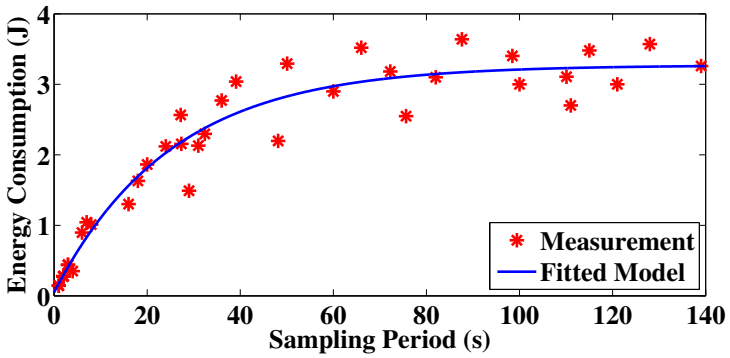


Figure 9: Phone's GPS module energy consumption measurements and the fitted model

7.1.3 Computations

The only computations that are eNav-specific and carried out continuously for every time slot (set to be 1s in our experiment) throughout the entire navigation are the car-idle and car-turning detections and the principal motion calculation. Both operations involve computing simple statistics (e.g., $mean$) of the accelerometer data segment (of size 10×3). The principal acceleration calculation then only involves taking the dot product of two 3D vectors. The idle/turning detections involve decision tree prediction computations, where the decision trees are usually of depth < 10. Therefore,

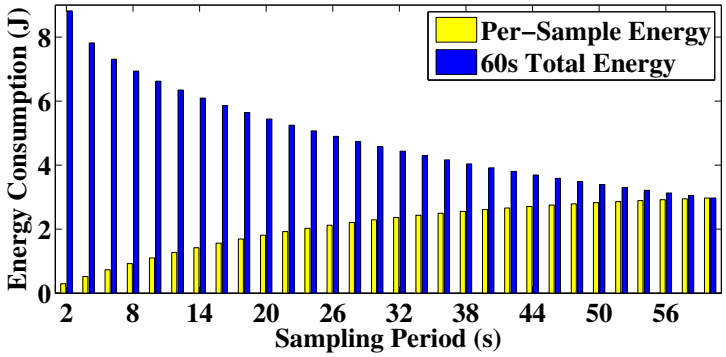


Figure 10: Phone's GPS energy consumption trends with increasing sampling period: per-sample energy vs total energy for 1min of sampling

each such detection involves comparing a pair of numbers at most 10 times. All these computations are lightweight, and their energy consumption as measured are negligible compared to the sensors and the GPS.

The training of the decision tree and the estimation of the principal motion vector using PCA are carried out once within the first minute of the trip instead of continuously throughout the entire navigation. And the computation energy consumption is negligible compared to the navigation's total energy consumption throughout the entire trip.

7.1.4 Energy Model Verification

We validated our energy models using data collected from about 300 km of driving traces, with two phones placed in the car under similar setting (both on the seat). One phone ran eNav and performed navigation, while the other phone simply logged the GPS and accelerometer data. The energy consumption of eNav during the navigation was recorded, and the data collected from the other phone was used to simulate the running of eNav and predict the energy consumption by applying our energy models. We observed that, over the aggregated driving trip data, the computed value remains within an $\pm 5\%$ error window of the measured energy consumption.

7.2 Navigation User Study

We evaluated the energy saving and navigation quality of eNav through a user study. We recruited 33 external (non-author) volunteer participants (from multiple departments of the university; of both genders; ages ranging from 20s to 40s), gave them Galaxy Nexus Android phones with our eNav prototype app installed. We explained to all participants beforehand that eNav was designed and built around the goal of energy efficiency without compromising navigation correctness, and asked them to use eNav on navigation trips, during which they could place the phones however they liked in their cars. The users were also told that eNav would by default keep the phone

screen off, but they could turn on the display at any point they felt like to. A total of over 3000 km of navigation driving trips were logged, including various road / traffic / weather conditions (urban, rural; rush hour, non rush hour; daytime, nighttime; sunny, rainy, snowy). Users were given randomly chosen source-destination navigation trips to drive on, where we made sure the destinations had not been previously visited by the users (otherwise the users would not need to use navigation systems in the first place).

7.2.1 Energy Savings

As indicated by our survey results, the majority of people find it acceptable to rely on voice guidance during navigation, and most people are willing to even completely turn off the phone screen during navigation to preserve phone battery when running low. Therefore, we consider the following as the baseline navigation strategy: sampling GPS constantly at 1Hz and having the phone display turned off during the navigation. We report the energy savings of eNav with and without the enhancement modes enabled, as compared to the base strategy. Results are collected in Table 4. As seen, compared to the base navigation strategy, the basic eNav scheme reduces the navigation energy consumption by about 65%. Enabling the idle and turning detection enhancement modules each introduces an additional $5 \sim 8\%$ energy saving. And finally, with both detection modules enabled, eNav cuts down the navigation energy by nearly 80%.

	Energy Savings (%)
eNav with Idle & Turn Detections	78.37
eNav with Turn Detection	73.42
eNav with Idle Detection	70.59
Basic eNav	65.64

Table 4: Energy savings of various navigation schemes as compared to the baseline navigation strategy

We also take a look at the energy saving breakdowns as eNav (with both enhancement modules enabled) operates on road segments of different lengths. As shown in Fig. 11, longer segments lead to higher energy savings. This is expected because a car, as being driven on a longer segment, remains "far away" from the next waypoint (and thus operates under low accuracy navigation mode) a higher percentage of time than on shorter segments. Inspecting the driving traces reveals that few long-distance highway driving took place in our experiment. Therefore, even though already observing over 80% energy saving from our experiment, we expect the energy saving to be even higher for navigation trips that involve significant portions of long-distance highway drivings, spanning tens or even hundreds of kilometers.

Note that even though eNav achieves lower energy savings on shorter segments than longer ones, it does not mean eNav navigation in urban settings would necessarily lead to much poorer energy saving results. This is because when navigation services compute routes, they rarely choose zigzag-shaped routes over simpler ones with less turnings; We observed this from using Google Map (which we currently use as the routing engine for our eNav implementation) to compute navigation routes for a large

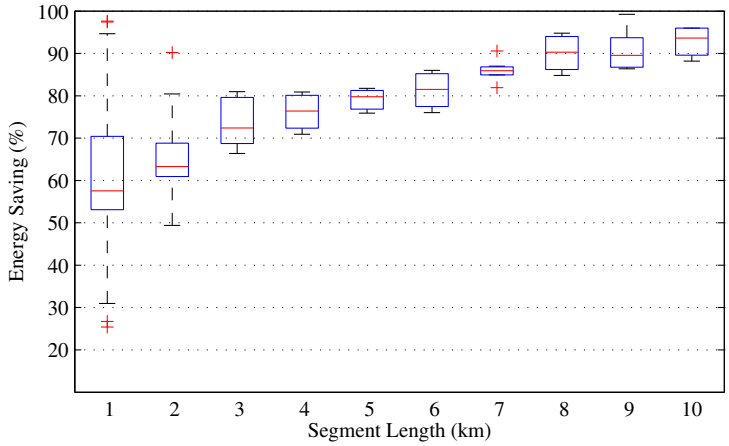


Figure 11: Energy saving distributions over different road segment lengths (the edges of the box are the 25th and 75th percentiles; the whiskers extend to the most extremes; outliers are individually marked)

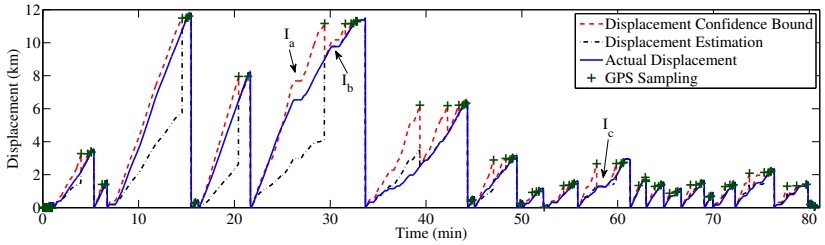


Figure 12: A complete navigation session for an entire trip using eNav with both car-idle and turn detection modes enabled. Every single segment's initial displacement value is set to 0 for ease and clarity of illustration.

number of randomly selected pairs of locations within urban areas. We tried this in Urbana-Champaign IL, New York City, and Seattle WA; The navigation routes computed rarely ($< 1\%$) contains more than 6 waypoints.

7.2.2 Navigation Quality

By navigation quality we refer to the ability of the navigation system to successfully deliver the navigation notification at or before the critical notification time. Traditional navigation applications are able to provide the highest possible navigation quality as they have constant access to high-accuracy location information. eNav, on the other hand, does not have the spot-on location information of the car at all times. Thus, it is natural to question whether navigation quality is impacted.

The driving trace data collected from our 33-user deployment study revealed that

eNav *never* missed a single delivery of navigation notification throughout the study. This means eNav is able to save energy during navigation without sacrificing quality as compared to traditional navigation apps. To compare to a traditional navigation app that samples GPS at a constant rate, we ran the navigation component of eNav at a constant rate and reduced the rate until its energy matched that of eNav's adaptive GPS sampling. The sampling period had to be increased from 1s to 83s. We then simulated running navigation using this low GPS sampling rate on our collected driving traces, and determine the timing of waypoint notifications. The experiment showed that only 16.8% of all way-points notifications were then delivered on time. Hence, the constant sampling rate service was unusable.

To more directly illustrate how eNav navigation works, Fig. 12 shows a complete example navigation session for an entire trip, which consists of both relatively long (around 10 km) and short segments (a couple of kilometers or about a few hundred meters). As seen, eNav samples the GPS very sparsely, and only when high location accuracy is needed. The traces for both the actual displacement estimation (computed using dead-reckoning, for estimating the car's location) and the displacement confidence interval (maintained for determining when GPS should be used) are shown. Car-idle in the middle of a segment can also be observed, for example, the I_a, I_b, and I_c as annotated in the figure. At these times eNav successfully detected that the car was not moving, reset its motion estimation, and prevented the unnecessary increase of the displacement estimation and the error confidence bound. The phenomenon that longer segments lead to higher energy savings is also evident.

7.2.3 End-to-End User Experience

We now discuss users' actual end-to-end experiences as they used eNav for navigations. Specifically, we look at their actual usage patterns, their real-life energy savings, and essentially how they felt about using eNav for navigation.

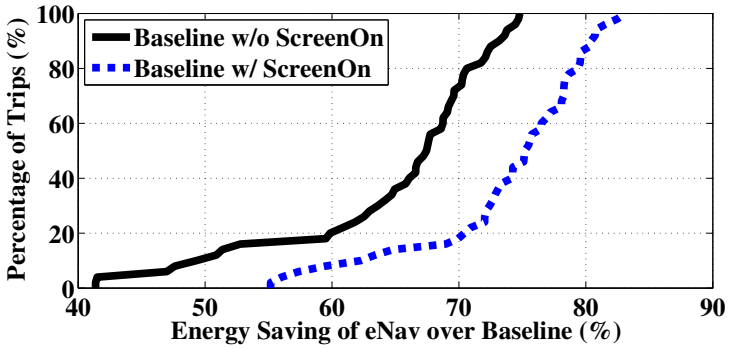


Figure 13: The CDF of energy savings of eNav over the base strategy in real uses

From our experiment data, we observed that about 68% of users never turned on the phone screen during navigations. Most turning-screen-on events occurred near waypoints, and were largely clustered around the destinations of the trips. The to-

tal screen-on time remained below 2% of the entire trip duration, except for one 6% occurrence, from a single user who, in particular, was an inexperienced driver and recently got a driver's license. The empirical CDFs of eNav's energy savings (over the aforementioned baseline navigation strategy of constantly sampling the GPS at 1Hz and also keeping the phone screen off) are shown in Fig. 13.

As users completed our study, we conducted a simple and informal exit interview to ask them about their general experience regarding using eNav for navigation in their own words. The most frequent comments we received were similar to the following, "*It's hard to tell the difference between your service and real GPS*," which we think well summarized eNav's user experience, and was as our original design goal indeed.

8 Discussion

The paper presented and evaluated eNav, a new navigation service that saves phone battery. A few elements of this work are worth highlighting, together with several limitations.

First, it was interesting for the authors to observe that technical affordances sometimes diverge from user preferences; even though the car is such an energy-rich technical artifact, and even though many individuals are in the habit of carrying phone chargers in their cars, those individuals (according to our survey) still want an energy saving mode on their phone-based navigation application. In fact a statistically significant correlation was found between using the phone charger in a car and wanting eNav. The survey suggests that a more careful market study might be in order for this type of software. The survey results may have been biased by the fact that all responders were CrowdFlower users. Nevertheless, the survey does suggest that at least a niche market exists for eNav is some segment of the society, represented by the survey takers.

Second, one should keep in mind that the intended contribution of this paper is to investigate the energy savings possible (thanks to dead-reckoning and adaptive GPS sampling) while performing location sensing in the context of vehicular navigation. The service exports many configuration parameters whose settings are best tuned through a more rigorous usability study. For example, (i) how big should the displacement confidence window be, (ii) how much warning time is needed for the driver before waypoints, and (iii) what type of user interface should be put in place to toggle between standard and energy saving navigation modes, are questions we deem out of scope in this paper. This paper focuses on evaluating the feasibility of low-power location sensing in the context of vehicular navigation. The exact numbers will differ with technology and usability settings, but the paper offers evidence that the approach can bring non-trivial savings.

Finally, the bulk of the paper focused on explaining a few of the design decisions by evaluating the alternatives in our design space and observing their empirical pros and cons. For example, why did we not use vendor's versions of dead reckoning services? Why did we decide on adaptive GPS sampling in lieu of uniformly decreasing GPS sampling rate? What enhancements improved location estimation accuracy? etc. Many more questions remain, that relate more to understanding different aspects of service performance. For example, how would savings compare in different cities such as

Los Angeles (lots of waiting, heavy traffic), Salt Lake City (large streets, perfect grid, little traffic), San Francisco (hilly streets, some traffic), and Ann Arbor (a small college town)? How do savings depend on the type of commute (e.g., freeways versus surface streets)? How do they depend on driver settings (e.g., waypoint warning time)? The possibilities are exponential. Such a detailed controlled study requires a much larger-scale deployment, which is beyond the means of the authors of this paper. The authors hope that the paper might encourage followups of more means to undertake such a more detailed investigation. The authors are currently working on releasing the application on Google Play to obtain more insights from its open use.

9 Related Work

There has been a fair amount of work studying mobile-based location sensing [6, 20, 22, 24, 28, 40]. The costly nature of GPS sensing on phones is widely recognized [16, 20, 22, 26, 41, 42]. Several approaches are then proposed to address this energy problem. For example, static [22] and adaptive duty cycle [4, 16, 28, 38] methods are studied; phone's on-board sensors (e.g., accelerometer) are also used to trigger GPS sampling [20, 24]; network-based (e.g., cellular, WiFi) methods [14, 16, 31] or other multi-modal-based approaches [11, 15, 35] are also studied as replacements for GPS as localization sources. These above work generally aims at balancing the trade-off between the *overall* location sensing accuracy and the energy consumption. eNav is different in that it specifically targets the vehicular navigation scenario, for which we do *not* care about the overall location accuracy; rather, only at a few discrete waypoints along the route do we really ask for spot-on localization. This unique nature of navigation opens up a whole new range of energy saving opportunities, which eNav exploits. This notion of variable localization accuracy requirement has also been studied in previous work [17], which, however, specifically considers navigation as not being an applicable scenario.

The problems of navigation have been studied, but only mainly for indoor and/or pedestrian scenarios [3, 5, 9, 35], which have natures of motion and localization opportunities that are completely different from driving scenarios.

Vehicles have been a target for several recent smartphone-based sensing systems [2, 18, 36, 39] that sense the dynamics of various aspects of people (e.g., driver phone use) as well as the vehicles themselves (e.g., speed variance). Our eNav system explores the vehicular sensing scenario to enable low-power navigation service.

For vehicular navigation, multiple work exists that try to combine GPS and motion sensors (accelerometer, gyroscope, etc.) [10,32,34]. However, they focus on improving the vehicle location tracking accuracy by building dedicated systems that use motion sensor data as supplements to GPS sensing. eNav, on the other hand, embraces a completely different design principle in approaching the navigation problem: the entire system is based on off-the-shelf mobile phones, and focuses on energy efficiency by trying to replace GPS accurate location sensing with dead-reckoning-based rough location estimation as much as possible, without compromising navigation quality.

10 Conclusion

In this paper we present the design, implementation, and evaluation of eNav, an smartphone-based energy-efficient vehicular navigation system. We take advantage of the mobile phone's on-board low-power accelerometer to relieve the GPS sampling burden from real-time navigation tasks. The navigator behaves exactly the same as traditional navigation applications in terms of navigation quality. At the same, time eNav achieves around 80% energy savings, relieving users from having to carry extra phone charging instruments, and improving the phone user's experience.

References

[1] B. Barshan and H. F. Durrant-Whyte. Inertial navigation systems for mobile robots. *Robotics and Automation, IEEE Transactions on*, 11(3):328–342, 1995.

[2] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R. P. Martin, J. Yang, and Y. Chen. Tracking vehicular speed variations by warping mobile phone signal strengths. In *PerCom*, 2011.

[3] D.-K. Cho, M. Mun, U. Lee, W. J. Kaiser, and M. Gerla. Autogait: A mobile platform that accurately estimates the distance walked. In *PerCom*, 2010.

[4] Y. Chon, E. Talipov, H. Shin, and H. Cha. Mobility prediction-based smartphone energy optimization for everyday location monitoring. In *SenSys*, 2011.

[5] I. Constandache, R. R. Choudhury, and I. Rhee. Towards mobile phone localization without war-driving. In *INFOCOM*, 2010.

[6] I. Constandache, S. Gaonkar, M. Sayler, R. R. Choudhury, and L. Cox. Enloc: Energy-efficient localization for mobile phones. In *INFOCOM*, 2009.

[7] CrowdFlower Inc. `http://www.crowdflower.com/`, 2013.

[8] Electronista. Garmin profits drop as hardware sales continue decline `http://goo.gl/8YVH8p`, 2013.

[9] Y. Gao, J. Niu, R. Zhou, and G. Xing. Zifind: Exploiting cross-technology interference signatures for energy-efficient indoor localization. In *INFOCOM*, 2013.

[10] M. S. Grewal, L. R. Weill, and A. P. Andrews. *Global positioning systems, inertial navigation, and integration*. Wiley-Interscience, 2007.

[11] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar. Autowitness: Locating and tracking stolen property while tolerating gps and radio outages. In *SenSys*, 2010.

[12] InvenSense. `http://www.invensense.com/`, 2013.

[13] I. Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[14] A. J. Khan, V. Ranjan, T.-T. Luong, R. Balan, and A. Misra. Experiences with performance tradeoffs in practical, continuous indoor localization. In *WoWMoM*, 2013.

[15] D. H. Kim, Y. Kim, D. Estrin, and M. B. Srivastava. Sensloc: sensing everyday places and paths using less energy. In *SenSys*, 2010.

[16] M. B. Kjærgaard, J. Langdal, T. Godsk, and T. Toftkjær. Entracked: energy-efficient robust position tracking for mobile devices. In *Mobisys*, 2009.

[17] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao. Energy-accuracy aware localization for mobile devices. In *MobiSys*, 2010.
[18] J. Lindqvist and J. Hong. Undistracted driving: A mobile phone that doesn't distract. In *HotMobile*, 2011.
[19] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. Loureiro, and Q. Wang. Energy efficient gps sensing with cloud offloading. In *SenSys*, 2012.
[20] H. Lu, J. Yang, Z. Liu, N. Lane, T. Choudhury, and A. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In *SenSys*, 2010.
[21] G. Milette and A. Stroud. *Professional Android Sensor Programming*. Wrox, 2012.
[22] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell. Sensing meets mobile social networks: the design, implementation and evaluation of the cenceme application. In *SenSys*, 2008.
[23] P. Misra, W. Hu, Y. Jin, J. Liu, A. S. de Paula, N. Wirström, and T. Voigt. Energy efficient gps acquisition with sparse-gps. 2014.
[24] P. Mohan, V. N. Padmanabhan, and R. Ramjee. Nericell: rich monitoring of road and traffic conditions using mobile smartphones. In *SenSys*, 2008.
[25] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda. Peir, the personal environmental impact report, as a platform for participatory sensing systems research. In *Mobisys*, 2009.
[26] S. Nath. Ace: exploiting correlation for energy-efficient and continuous context sensing. In *Mobisys*, 2012.
[27] OpenStreetMap. `http://www.openstreetmap.org/`, 2013.
[28] J. Paek, J. Kim, and R. Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In *Mobisys*, 2010.
[29] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan. Energy-efficient positioning for smartphones using cell-id sequence matching. In *MobiSys*, 2011.
[30] J. R. Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.
[31] S. Sen, J. Lee, K.-H. Kim, and P. Congdon. Avoiding multipath to revive inbuilding wifi localization. In *MobiSys*, 2013.
[32] I. Skog. A low-cost gps aided inertial navigation system for vehicular applications. *KTH Signals Sensors and Systems*, page 49, 2005.
[33] S. H. Stovall. Basic inertial navigation. *Naval Air Warfare Center Weapons Division*, 1997.
[34] M. L. G. Thoone and H. P. M. Krukkert. Adaptive inertial vehicle navigation system, Mar. 7 1990. EP Patent 0,181,012.
[35] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury. No need to war-drive: Unsupervised indoor localization. In *MobiSys*, 2012.
[36] Y. Wang, J. Yang, H. Liu, Y. Chen, M. Gruteser, and R. P. Martin. Sensing vehicle dynamics for determining driver phone use. In *MobiSys*, 2013.
[37] Wikispeedia. `http://www.wikispeedia.org/`, 2013.
[38] C.-L. Wu, Y.-T. Huang, C.-L. Wu, H.-h. Chu, P. Huang, and L.-J. Chen. An adaptive duty-cycle scheme for gps scheduling in mobile location sensing applications. *PhoneSense*, 2011.
[39] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen,

M. Gruteser, and R. P. Martin. Detecting driver phone use leveraging car speakers. In *MobiCom*, 2011.

[40] C.-w. You, Y.-C. Chen, J.-R. Chiang, P. Huang, H.-h. Chu, and S.-Y. Lau. Sensor-enhanced mobility prediction for energy-efficient localization. In *SECON*, 2006.

[41] A. Zhan, M. Chang, Y. Chen, and A. Terzis. Accurate caloric expenditure of bicyclists using cellphones. In *MobiSys*, 2012.

[42] Z. Zhuang, K. Kim, and J. Singh. Improving energy efficiency of location sensing on smartphones. In *Mobisys*, 2010.